BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a generally diagrammatic view of a single track of a pair of railroad tracks equipped with a heater for melting ice and snow.

Fig. 2 is a generally diagrammatic view similar to Fig. 1 showing how the prior art is modified with my invention.

Fig. 3 is a block diagram of a remote module for activating and deactivating railroad switch heaters.

Fig. 4 is a block diagram generally showing layout of my invention.

Figs. ~~Fig.~~ 5 and 5a together form ~~is~~ a block diagram showing architecture of my invention.

Fig. 6 is a screen shot, by way of example, of a main menu of a railroad switch heater system of the instant invention.

Fig. 6a is a screen shot, by way of example, of a railroad switch heater yard configuration page.

Fig. 6b is a screen shot, by way of example, of a menu for configuring discrete heaters of the system of instant invention.

Fig. 6c is a screen shot, by way of example, of a menu for adding railroad switchyards to the system of the instant invention.

Fig. 6d is a screen shot, by way of example, of a menu for adding additional discrete heaters to the system of the instant invention.

Fig. 6e is a screen shot, by way of example, of a menu for selecting railroad switchyards in the system of the instant invention.

Fig. 7 is a screen shot, by way of example, of a control page for controlling

railroad switch heaters in a railroad yard and for displaying status and alarm information.

Fig. 7a is a screen shot, by way of example, of a control page for controlling discrete railroad switch heaters.

Fig. 7b is a screen shot, by way of example, of a control page showing a pop-up indication of an alarm.

Fig. 7c is a screen shot, by way of example, of a configuration menu for managing accounts (end user companies) of the instant invention.

Fig. 7d is a screen shot, by way of example, of a configuration page for adding accounts (end user companies) of the instant invention.

Fig. 8 is a software flow diagram, by way of example, of an initialization process of software of the instant invention.

Fig. 9 is a software flow diagram illustrating, by way of example, the process by which pages (MIN numbers) are passed to a gateway server.

Figs. ~~Fig.~~ 9a and 9aa together form ~~is~~ a software flow diagram illustrating, by way of example, a process of the instant invention for receiving registrations from remote modules.

Fig. 9b is a software flow diagram illustrating, by way of example, a process of the instant invention by which a gateway messenger functions.

Fig. 9c is a software flow diagram illustrating, by way of example, a process of the instant invention for tracing transactions.

Fig. 9d is a software flow diagram illustrating, by way of example, operation of a gateway communicator of the instant invention.

Fig. 9e is a software flow diagram illustrating, by way of example, another process of the instant invention by which the gateway communicator functions.

Fig. 9f is a software flow diagram illustrating, by way of example, a process of the instant invention by which a batch of MIN numbers are registered.

Fig. 9g is a software flow diagram illustrating, by way of example, a process of the instant invention adding or deleting a MIN number.

Please substitute the following paragraph for the paragraph beginning at Pg. 22 line 2.

More specifically, and referring to the block diagram of Figs. ~~Fig.~~ 5 and 5a, which may be fitted together at lines A – A and B – B to form a single block diagram, the control center 43 of Fig. 4 is shown. Here, graphical user interface 72 may include any operating system, such as Windows 2000™ or Windows NT™. Other operating systems, such as LINUX™ and UNIX™ may also be used as would be determined by a skilled programmer. Any browser, such as Internet Explorer™, Netscape™, Eudora™, Mozilla™ or another as determined to be appropriate by a skilled programmer may be used. As stated, interface 72 may be in a client company computer, in addition to an interface 72 in the service company system. A web server or general-purpose computers 70 generally configured as shown and described may be in a client company location. Further, web server 70 and remote modules server 74 may be configured as software modules that may be installed on a client company's computer system. Further yet, a plurality of remote module server software modules and web server modules may be installed in one or more computer servers of my service company.

Please substitute the following paragraph for the paragraph beginning at Pg. 26 line 6 of the specification as originally filed.

Event data received by event dispatcher 96 is generated by event generator 118, which receives inputs from health center 119, registration handler 106, diagnosis engine 114 and page issuer 92. With respect to health center 119, any failure with respect to overall operation of the system and errors that are returned will elicit an alarm by health center 119, which alarms and errors being passed to event generator 118. With respect to commands and requests, page issuer 119 provides a return indication to event generator 118 that the page containing one or more commands or requests was successfully sent. If the page was not successfully sent, an acknowledgement signal from the gateway server is not received and the command or request is not deleted from hash table 120. This results in two attempts to resend the page, after which an error is generated. A received acknowledgement response to sending a page to a remote unit is passed to gateway communicator 116, and subsequently to gateway server messenger 110. Messenger 110 provides the acknowledgement signal in the form of a registration, and places the registration in registration queue 112. From there, registration handler 106 periodically polls registration queue 112, and picks up the registration and processes the registration as shown in Figs. ~~Fig.~~ 9a and 9aa as will be described.

Please substitute the following paragraph for the paragraph beginning at Pg. 34 line 15 of the specification as originally filed.

A series of flowcharts will now be described, with functions of these flowcharts being generally related to remote server 74 in the block diagram of Figs. ~~Fig.~~ 5 and 5a and the screen images of Fig. 7 - 7e. Here, Fig. 8 shows an initialization sequence. First, at box 200, the command queue, event queue, and transaction hash table 120 (Figs. ~~Fig.~~ 5, 5a), labeled Slist, and registration queue are initialized, and where appropriate populated with default values. Next, at box 202 the autoresetevent signal, GWSregistration, GWSregisterMINack signal and GWShealthack message are initialized. At box 204 the GW communicator is initialized to establish the socket connection to the gateway server. At box 206 an inquiry is made as to whether the socket connection to the gateway server was successful, and if unsuccessful, then the program returns a FAIL signal and exits at box 208. If the connection was successful, then at box 210 the gateway server messenger and gateway server health checker are initialized. At box 212 the gateway server messenger thread is started, allowing the gateway server messenger to run at box 214. At box 216 the transaction tracer thread is started, allowing the transaction tracer to run at box 218. At box 220 the gateway server health checker thread is started, allowing the gateway server health checker to run at box 222. At box 224 the implicit register MIN, i.e MIN register 122 (Figs. ~~Fig.~~ 5, 5a) retrieves all MIN numbers for the remote modules from database 78 and passes them via the gateway communicator 116 to hash table 150 in gateway server 76. At box 226 the gateway server registration handler thread is started, allowing the

registration handler to run at box 228. At box 230 the gateway server page issuer thread is started, allowing the page issuer to run at box 232. It should be noted that the threads of boxes 214, 218, 222, 228 and 232 run as endless loops, i.e when they reach the end, as shown on their respective flowcharts, they loop back to the beginning and run again.

Please substitute the following paragraph for the paragraph beginning at Pg. 35 line 18 of the specification as originally filed.

Fig. 9 is a flowchart of one method by which pages may be issued by the page issuer thread 232 initialized in Fig. 8. At box 250 the query is made as to whether the command queue 88 (Figs. ~~Fig.~~ 5, 5a) for issuing commands to the heater remote modules is empty or if commands are present in the queue. In the instance where the command queue is empty, the program simply loops back to ask the question again. If the command queue is not empty, as indicated by a " NO" answer, meaning that at least one command is in the queue, such as a command to energize or deenergize a remote heater module, to read a meter or get status information from a remote module, then the command request is retrieved by web messenger 89 from the queue at box 252. At box 254 the question is posed as to what type of command has been retrieved. If the command is an individual command, i.e. a command to a discrete remote module, such as to energize or deenergize a specific railroad switch heater or switch a capacitor bank for a utility system, then the program proceeds to box 256 where the question is asked as to whether the same MIN is in the transaction hash table 120, meaning that the remote module is busy processing a previously-issued command. If the MIN is found in the status list of hash table 120, then the answer at box 256 is YES, meaning that the action is in progress. Here, while the action at the remote module takes little time to accomplish, sending the page and receiving an associated registration may require a minute or more. Thus, at box 258 a report is generated via event generator 118 (Fig. 5) indicating that the requested MIN is already being processed, with this report being shown in window 202 (Fig. 7). Similarly,

where a group of MINs are requested, as where all switch heaters are to be energized, and one or more are already in process, then corresponding reports are generated through event generator 118. If the command type is a register MIN (box 262), as where a new remote module is added to the system, a MIN number is added to database 78 for the new remote module. In this instance, the new MIN number is added to MIN register 100, which in turn provides it to MIN register 122 where it is passed via gateway communicator 116 and socket manager 132 to hash table 150 of gateway server 76. Where the answer at box 256 is NO, meaning that the transactions are not in progress, then the program falls through to box 264 where the request or requests is/are inserted into the transaction hash table 120. This causes, at box 266, a "PAGE ISSUE" to be initiated that cumulates in the issuance of a page containing the command MIN. At box 268 the command MIN is obtained along with switching center information for the requested page by page issuer 92, and at box 270 the page is issued to gateway server 76 via gateway communicator 116. At box 272 the query is made as to whether the page was successfully issued, as by reception of an acknowledgement signal from the cellular system, and if so then at box 274 "ISSUE SUCCESS" is associated with the respective MIN in hash table 120. At box 278 "ISSUE COMMAND SUCCESSFUL" is reported to web server 70 through event generator 118, which reports a successful issue of the command in box 202 (Fig. 7), and the program exits. If the issued command was not successful at box 232, then at box 280 "ISSUE FAIL" is associated with the MIN number in hash table 120 and at box 282 error

information is saved in an exception log table in database 78. In the instance of a major failure, the failure message may be provided as a pop-of window in Fig. 7, or where the failure is a minor failure the failure message may be associated with a respective one of buttons 200 or provided in status window 202.

Please substitute the following paragraph for the paragraph beginning at Pg. 37 line 23.

Figs. ~~Fig.~~ 9a and 9aa, which may be connected to form a single flowchart by matching lines A – A, B – B and C – C illustrate ~~illustrates~~, by way of example, one possible logic flow for handling registrations, i.e. the registration handler thread 228 of Fig. 8. Generally, this logic flow describes how registration data is obtained from a registration queue, the data being parsed and reports generated containing, where appropriate, an error message, ESN data, status information and the status, alarm or other message saved in database 78. More specifically, at box 290 the registration is buffered in registration queue 112, and gateway server 76 notifies registration handler 106 by a synchronic signal that a registration is waiting to be picked up, at which point the registration message is obtained by gateway server messenger 110 at box 292. At box 294 the query is posed as to whether or not the message is a registration message or an error message. In the instance where the message is a registration error message, then at box 296 the event "ALARM REPORT" is reported to web server 70 via event generator 118. As described, the error message may be displayed in status window 200 (Fig. 7), a pop-up window or be associated with an icon. At box 298 the inquiry is posed as to whether or not the MIN number is found in hash table 120. If so, then at box 300 "TRANSACTION FAIL" is reported to web server 70 via event generator 118 and an event is reported, as by displaying a message in status window 202.

A complete listing of the claims and their status follows: